Stylised Image Generation From Deep Neural Networks

1st Yameng Peng
School of Science
RMIT University
Melbourne, Australia
s3712236@student.rmit.edu.au

2nd Vic Ciesielski
School of Science
RMIT University
Melbourne, Australia
vic.ciesielski@rmit.edu.au

***Abstract*—The purpose of convolutional neural networks is usually image classification but there are increasing studies attempting to reverse this common purpose in order to generate images. One of the most promising research directions is style transfer. This involves rendering the overall texture of an image into an artistic style. There are two common approaches in this field, which are feature representation based methods and generative adversarial network(GAN) based methods. In this paper, we focus on GAN based methods. We observed that most variants of GAN usually need paired data in order to generate the desired result, the training costs are very heavy and the quality of the result is not guaranteed. We propose an improved architecture for generative adversarial models for multi-style rendering. A new loss function configuration enables learning from unpaired data and generation of stylized images with specific artistic styles from normal photographs. A weighted combination of loss functions can control the trade-off between style and content of a stylized image.**

***Index Terms*—AI-generated art, Style transfer, Deep neural network, Generative adversarial network.**

I. Introduction

Creativity is usually considered as a special gift of human beings, the oldest artworks such as cave paintings can be traced back to ten thousand years ago [1]. Due to the thousands of years of practice, humans have mastered the use of various artistic styles and contents to create remarkable paintings, but the mechanism of creativity itself is still a mystery. With the progress of modern computer science, researchers have started to use different approaches to create interesting and eye-catching artworks on machines, such as genetic algorithms [2], [3]. In this research, we will focus on deep learning based image generation methods.

In earlier days, the artificial neural network is basically a mathematical function which maps a set of input values to output values. When we want to handle an image classification task, there may be millions of pixels within an image. This leads to huge computation pressure and the network is hard to train. To deal with this problem, Hinton et al. proposed distributed representation and suggested that we should extract features from an image and train the network based on feature representations [4]. This idea spawned the convolutional neural network (CNN) [5], its major components are convolution layers, pooling layers, and fully connected layers. The convolution kernel is a filter which is able to extract visual features from an image as the feature maps.

Convolutional neural networks have achieved excellent results on visual processing such as image classification and object detection [6]. The network takes an image as input and converts it into feature maps and outputs a probability distribution of classes. The probability distribution is a 1-dimensional array, its length is the number of classes for a certain classification task [7]. It arouses our curiosity and research interest to reverse its regular process and purpose and to make the network generate interesting images rather than classifying images.

A. Research Questions

During the research on related work, we replicated several studies. Several issues were identified in our early experiments. First, the vanilla GAN cannot directly take an image as input [8]. Second, its variants are usually unable to reconstruct delicate contents from a large dimension artistic painting [9]. Third, the network requires paired training data, but the data may not preexist in some circumstances such as stylized image generation. Based on those observations and our original intention, we want to propose a new approach to address these issues. The research questions we aim to answer in this paper are:

1) How can we use deep neural networks to generate images with particular artistic styles from normal photographs?
2) How can we control the trade-off between style and content of stylized images generated from the above networks?

In order to answer the above research questions, this paper is organized as follows: Section 3 describes how we use an alternative loss function to replace the binary cross-entropy from the vanilla GAN and add an auxiliary classifier to the model. Then, we propose a supplementary loss function and an encoder/decoder structure to the generator network in order to handle the style rendering. In the same section, we describe the structure of the proposed model, implementations of the model and experimental setup along with sample stylized images and their analysis. Section 4 describes three factors which can control the trade-off between style and content of a stylized

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

image. We show some representative outputs generated from specific combinations of those factors. Section 5 gives our conclusions and suggestions for future work.

B. Research Contributions

This research has made the following contributions to the field of image generation from generative adversarial networks:

1) We have shown that a new loss function combination for the generative adversarial network is effective for style rendering.
2) We have shown that a weighted combination of 3 loss functions can control the trade-off between style and content of a stylized image.
3) We have shown that our proposed method is able to generate stylized images with multiple styles in a single GAN system using unpaired training data.

II. Related Work

There are two different research directions for the task of stylized image generation. The first direction is the feature representation based. It derives from studies of layer visualization. It looks for desired feature representations and reconstructs an image from a combination of desired feature representations. A representative of this kind of research is Google's Deep Dream [10], which is able to transfer the textures and colours from a style image to a content image [11], [12]. But the network in this approach is not learning how to generate images, only its feature representations are used to reconstruct images. A pre-trained network extracts feature representations from a content image and a style image and another optimization algorithm merges them to generate the final stylized image. Due to this, the whole feature extraction and optimization processes have to be executed from beginning in order to generate a stylized image, it is inefficient.

The second direction is the generative adversarial network (GAN) based [8]. This consists of two components which are the generator model and the discriminator model, both of them are artificial neural networks. The generator will generate fake samples from random noise inputs and aim to fool the discriminator. The job of the discriminator is to distinguish real samples and fake samples and provide feedback to the generator. Due to this competitive process, the generator is able to generate images which look similar to the real samples.

Following the idea of the generative adversarial network, there are many related studies came out. Mao et al. observed that the vanilla GAN was troubled by the vanishing gradients problem during the training process because it used sigmoid cross-entropy as the loss function. They replaced cross-entropy by a least-squares loss function in order to gain a more smooth and stable training process [13]. Radford et al. observed the success of CNN in supervised learning and they tried to bring the advantages of CNN into unsupervised learning field. They replaced the regular neurons of a vanilla GAN by convolutional neurons in order to make the networks learn from feature representations [14]. The earlier works of GAN were limited to generating images with simple scenarios and clean backgrounds, such as handwritten characters. Otherwise, it was difficult to reconstruct clear contents. Odena et al. proposed an improved GAN model which contains an auxiliary classifier to overcome the issue. They added the categorical cross-entropy as the auxiliary classifier loss in order to use the class information [15]. The input data for the generator network of vanilla GAN and follow-up work are mostly random noise. Due to this mechanism, the generator networks of previous GANs cannot take an image as input. Pathak et al. proposed their context encoder model which is a variant of GAN, but the generator network contains an encoder-decoder structure in order to take an image as input [16]. Although the encoder and decoder structure achieved impressive results, Li et al. observed that it was limited to rather small dimension images and fidelity in detail. They proposed the PatchGAN model to overcome this issue [17]. Based on the success of previous work on GANs, Isola et al. published their conditional adversarial network for image-to-image translation in 2017. The discriminator learns to distinguish real/fake from paired samples and pushes the generator to generate more realistic images. This model was able to translate the input image to another style, such as from semantic label maps to realistic photographs [18].

III. Stylized Image Generation via a Generative Adversarial Model

A. Alternative Loss Functions for Generative Adversarial Network

Goodfellow et al. summarized their generative adversarial network as Equation (1) which uses binary cross-entropy as the loss function [8]. The generator G takes an input x and tries to generate an image $G(x)$ which is similar to the target image set Y. The discriminator D tries to distinguish the real Y and generated image $G(x)$. The training strategy is designed to maximize the log-likelihood of discriminator D which is expected to correctly distinguish real image Y and generated image $G(x)$. Meanwhile, the generator G has to minimize the log-likelihood of D, fool the discriminator and make it believe that $G(x) \in Y$.

$$\underset{G}{min}\underset{D}{max} L(G, D) = E_{y \in Y}[logD(y)] + E_{x \in X}[log(1 - D(G(x)))] \tag{1}$$

The shortcoming of using binary cross-entropy as the loss function is that may lead to the vanishing gradients problem during the learning process. In order to overcome this problem and implement a patch-based mechanism later, we adopt the loss function of LSGAN [13]. It uses a least-squares loss function for the discriminator (Equation 2) which is able to provide higher quality images than the vanilla GAN, giving a stable and smooth learning process.

$$L(G, D) = E_{y \in Y}[(D(y) - 1)^2] + E_{x \in X}[D(G(x))^2] \tag{2}$$

B. Auxiliary Classifier For Style Classification

Our goal is to generate a stylized image with a particular artistic style, however the above model unable to learn the differences between multiple artistic genres. To deal with this limitation, we adopt an auxiliary classifier from ACGAN [15] which use categorical cross-entropy as the auxiliary loss (Equation 3) to identify the style label.

$$L_C = -E_{y,c \in Y}[logP(Style = c|y)] \tag{3}$$

$$L_Y = E_{y \in Y}[(D(y) - 1)^2] + E_{x \in X}[D(G(x))^2] \tag{4}$$

In our model, we combine L_Y and L_C together as the overall loss function (Equation 5). L_Y will indicate the similarity between a real sample from Y and a generated image. L_C will determine which artistic style the input sample belongs to. The auxiliary classifier has to predict the correct artistic style of a real sample from Y. At the same time, the style of a generated image should also be correctly predicted.

$$L(G, D) = L_Y + L_C \tag{5}$$

In order to make the generator able to take an image and target style label as input, we adopt the encoder and decoder structure from Context Encoder [16]. In this structure (Equation 6), the generator consists of three parts: An encoder EC, a label embedding layer LE and a decoder DC. The EC will downsample a given image x into feature representations and pass it to the LE. LE will do an element-wise embedding which binds the style label c into feature representations and passes it to the DC. DC will upsample the feature representations and reconstruct a stylized image. We modified the network from the regular convolutional block to the residual block. This makes the network easier to optimize and to obtain accuracy from a very deep network [19], [20].

$$G(x, c) = DC(LE(EC(x), c)) \tag{6}$$

C. Supplementary Loss Function For the Generator

There still exists a problem in the above loss function. It is missing an anchor of the input image from X. The generated image will more and more similar to the artistic image from Y and lose the content from X, as the network can only learn information about an artistic image from the current loss function. In order to do the style rendering and keep the content from the original input, we propose a supplementary loss L_X (Equation 7). It is an L1 loss which enables the generator to learn differences between the output $G(x, c)$ and the original input image x. By adding L_X to the generator loss (Equation 8), the generator will learn extra knowledge which the discriminator does not know.

$$L_X = E_{x \in X}[|G(x, c) - x|_1^1] \tag{7}$$

$$L(G) = L_Y + L_C + L_X \tag{8}$$

D. Structure Of the Proposed Model

Fig 1 shows the structure of our proposed GAN model. The generator contains three components which are: the encoder, label embedding layer and the decoder. The model takes a photograph and a target style label as input. The encoder will

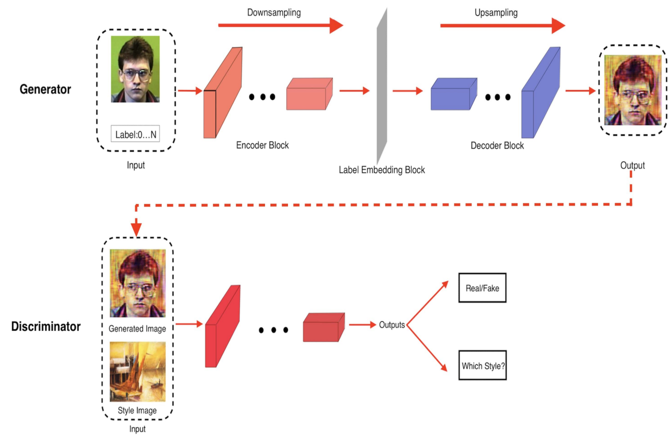


Fig. 1: Structure of the proposed model

downsample the input image into feature representations and pass it to the label embedding layer which will concatenate the target label with the feature representations and finally pass it to the encoder and reconstruct a stylized image as its output. Next, the generated image and real paintings will be input into the discriminator which will generate two outputs: a judgment of authenticity of the given image and its style class.

E. Model Implementations and Experimental Setup

We trained our network on the Google cloud platform, the virtual machine consists of a quad-core Intel CPU, 26 GB memory and an Nvidia Tesla P100 GPU. The content dataset contains approximately 4000 portrait photographs, the style dataset contains approximately 1500 artistic paintings with 3 genres which are Impressionism, Cubism, and Abstract (Fig 2), and we use 0, 1, 2 to indicate those styles respectively during the network training. We randomly select the 90% images from each dataset for training and use the remaining 10% for model validation, the maximum number of training iterations is set to 1000.

(a) Impressionism (b) Cubism (c) Abstract

Fig. 2: Sample Artistic Paintings

Table 1 and 2 describe the implementations of the generator and discriminator models of our proposed method. We adopt residual block in the encoder and decoder parts and also in each layer of the discriminator. The numbers of convolutional kernels of the encoder and decoder are almost mirrored. They are [32, 64, 128, 256, 512, 512, 512] and [512, 512, 256, 128,

Layer Type	**Feature Maps**	**Kernel Size**	**Stride**	**Instance Normalization**	**Activation**
Input	**3** x 128 x 128	-	-	False	-
Encoder	**32** x 64 x 64	4	2	True	ReLU
Encoder	**64** x 32 x 32	4	2	True	ReLU
Encoder	**128** x 16 x 16	4	2	True	ReLU
Encoder	**256** x 8 x 8	4	2	True	ReLU
Encoder	**512** x 4 x 4	4	2	True	ReLU
Encoder	**512** x 2 x 2	4	2	True	ReLU
Encoder	**512** x 1 x 1	4	2	True	ReLU
Label Embedding	**512** x 1 x 1	-	-	False	-
Decoder	**512** x 2 x 2	4	2	True	ReLU
Decoder	**512** x 4 x 4	4	2	True	ReLU
Decoder	**256** x 8 x 8	4	2	True	ReLU
Decoder	**128** x 16 x 16	4	2	True	ReLU
Decoder	**64** x 32 x 32	4	2	True	ReLU
Decoder	**32** x 64 x 64	4	2	True	ReLU
Output	**3** x 128 x 128	5	1	False	Tanh

TABLE I: Architecture of Generator Model

Layer Type	**Feature Maps**	**Kernel Size**	**Stride**	**Instance Normalization**	**Activation**
Input	**3** x 128 x 128	-	-	False	-
Convolution	**64** x 64 x 64	5	2	True	LeakyReLU
Convolution	**128** x 32 x 32	5	2	True	LeakyReLU
Convolution	**256** x 16 x 16	5	2	True	LeakyReLU
Convolution	**512** x 8 x 8	5	2	True	LeakyReLU
Output01	**1** x 8 x 8	5	1	False	-
Output02	**3**	-	-	False	Softmax

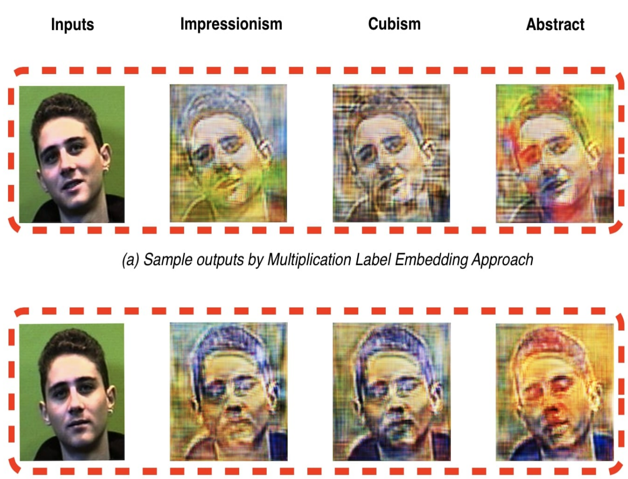
TABLE II: Architecture of Discriminator Model

64, 32]. We adopt instance normalization instead of regular batch normalization used by others [21]. Also, we use the rectified linear unit (ReLU) as the activation function except for the input and output layers. For the discriminator model, the number of convolutional kernels is [64, 128, 256, 512], we set the kernel size to 5 x 5 which is a relatively large kernel as we want to encourage the discriminator to focus on the global features. We adopt LeakyReLU instead of regular ReLU as our activation function. The discriminator has two output layers, *Output01* which generates a 1 x 8 x 8 patch for evaluating the real/fake image [18] and *Output02* which generates a probability distribution via softmax for classifying the style.

F. Sample Outputs and Analysis

We adopted two different label embedding approaches during this experiment: element-wise linear addition and element-wise multiplication. Sample outputs are shown in Fig 3. We observed that the generated images more obviously show style with multiplication label embedding. The specific style of generated images is more different from other styles (Fig 3a). The shortcoming is that content from the original input is more blurry than with the linear addition approach (Fig 3b). Overall, the generated images of both label embedding approaches roughly capture the overall feeling of each style class.

The first column shows the photograph sampled from the content dataset. The other positions show the stylized output. The multiplication label embedding approach encourages the label to be involved more in the stylizing process, the portrait becomes blurry compared to the original input. The linear addition label embedding approach makes the label a minor contributor to the stylizing process, the output maintains a relatively clear portrait.



(a) Sample outputs by Multiplication Label Embedding Approach

(b) Sample outputs by Linear Addition Embedding Approach

Fig. 3: Sample outputs by two different label embedding approaches

IV. Factors Controlling The Trade-off Between Style And Content

A. The Effect of Each Loss Function

Equation (9) shows the 3 loss functions involved in our proposed model, each one of them has a different influence on the generated image. Y represents the real artistic painting set, C represents the style classes of Y and X represents the content image set. Theoretically, L_Y will influence the overall feeling of the generated image such as global texture and colour gradient and guide the generated image to be similar to a real artistic painting from domain Y. L_C will magnify the differences between the generated images with different artistic styles, such as colours. L_X will try to retain a clear outline of the generated images in order to guarantee the content derived from X still recognizable.

$$\begin{cases} L_Y &= E_{y\in Y}[(D(y)-1)^2] + E_{x\in X}[D(G(x))^2] \\ L_C &= -E_{y,c\in Y}[logP(Style=c|y)] \\ L_X &= E_{x\in X}[|G(x,c)-x|_1^1] \end{cases} \tag{9}$$

B. The Factors For Controlling The Contribution of Each Loss Function

To precisely control the trade-off between style and content of a generated image, we propose three weights for above three loss functions, W_y, W_c and W_x. By adjusting the ratio of the loss weights, we are able to magnify the effect of a specific loss function described in Section 4.1. This will push the generated image in different directions from the aesthetic perspectives. Our final proposed loss functions for the discriminator and the generator shown in Equation (10).

$$\begin{cases} L(D) = L_Y + L_C \\ L(G) = W_y * L_Y + W_c * L_C + W_x * L_X \end{cases} \tag{10}$$

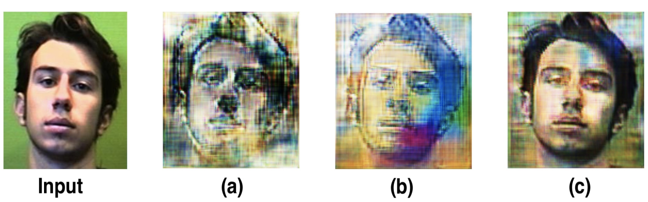


Fig. 4: Expected outputs from setting loss weights to extreme values

Fig 4 shows the expected outputs from setting the loss weights to extreme values, the input is a portrait photograph. When maximizing W_y (image a), the output should be blurry and global details similar to an artistic painting, including content from the input. When maximizing W_c (image b), the output should contain more intense colours compared to other cases. When maximizing W_x (image c), the output should retain a clear profile of the content from the input, but have relatively fine style rendering on the background.

C. Model Implementations and Experimental Setup

Based on the earlier work we have done, we adopt the same model architecture from Section 3 with the linear addition label embedding approach for the following experiments. In addition, we add loss weights into the model to explore different outputs. To determine the actual effects of each weight, we designed three extreme weight combinations. In combination 1, we set W_y and W_c to 1, W_x to 10, in order to magnify the effect of W_x. In combination 2, we set W_y and W_x to 1, W_c to 10, in order to magnify the effect of W_c. In combination 3, we set W_c and W_x to 1, W_y to 10, in order to magnify the effect of W_y. The following experiments were carried out on the same cloud platform, training datasets, and training strategies as the experiments we have done in Section 3.

D. Sample Outputs and Analysis

Sample outputs of the three combinations are shown in Fig 5, Fig 6 and Fig 7. We observed that combination 1 retains the clearest profile of the content, combination 2 shows the most intense colours and the differential between styles, combination 3 generates the most blurry outputs as the W_y guides the generator to focus on the global texture. The results are basically in line with the hypotheses we proposed in Section 4.2.

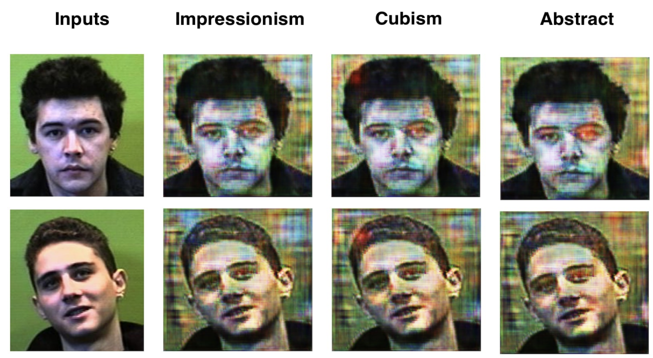


Fig. 5: Sample outputs by setting $W_y = 1, W_c = 1, W_x = 10$ In this extreme weight combination, W_x magnifies the contribution of L_x which makes the output maintain a relatively clear portrait and makes it similar to the original input.

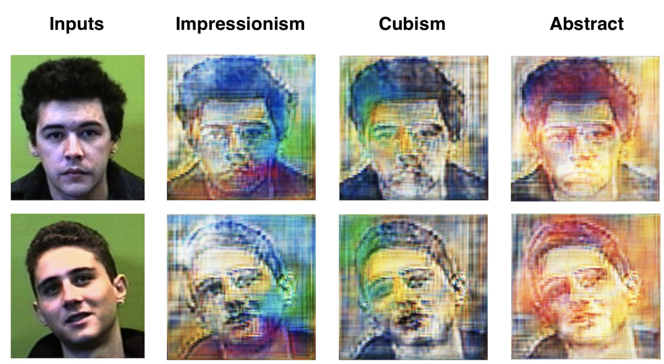


Fig. 6: Sample outputs by setting $W_y = 1, W_c = 10, W_x = 1$ In this extreme weight combination, W_c magnifies the contribution of L_c which makes the output present intense colours and a blurry portrait.

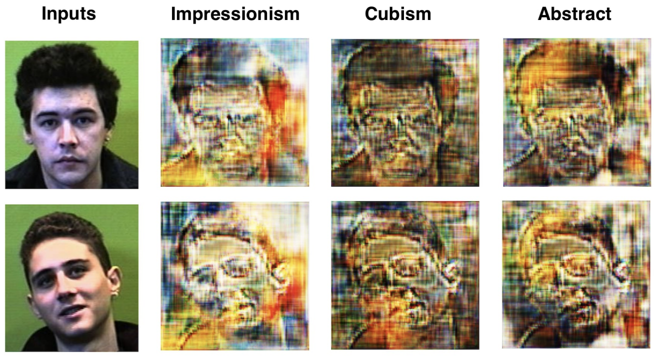


Fig. 7: Sample outputs by setting $W_y = 10, W_c = 1, W_x = 1$
In this extreme weight combination, W_y magnifies the contribution of L_y which makes the texture of generated image similar to the style image, but the portrait is almost unrecognizable.

E. Further Experiments

After verifying our hypothesis relating to factors which are able to control the trade-off between style and content, we implemented some fine-tuning of W_x, W_c, and W_y in order to generate output which we found attractive based on our personal aesthetic preferences. For example, the sample outputs in Fig 8 and 9, we fixed W_x to 10 to obtain a clear portrait in the output and W_c to 1 in order to avoid intense colours. We carried out an exploration of different values of weight combination, the weight combination $\{W_y = 2.5, W_c = 1, W_x = 10\}$ provided relatively balanced results which retain both clear portraits and eye-catching artistic styles (Fig 10).

Fig. 8: Sample outputs by setting up $W_y = 1.5, W_c = 1, W_x = 10$
In this weight combination, we increased W_y from 1 to 1.5 in order to bring more artistic feeling to the outputs.

Fig. 9: Sample outputs for $W_y = 3, W_c = 1, W_x = 10$
In this weight combination, we increased W_y from 1.5 to 3 in order to further increase the contribution of L_y.

V. Conclusion and Future Work

A. Conclusion

In this paper, we proposed a new combination of loss functions for the generative adversarial network in order to generate stylized images and weight combinations which enables us to control the trade-off between style and content.

Answers to Research Questions
Our first research question was:
How can we use deep neural networks to generate images with particular artistic styles from normal photographs?

To address this question, we proposed a new loss function combination in Section 3. The combination contains a least-squares loss function as the adversarial loss L_y, a categorical cross-entropy as the auxiliary loss L_c for the style classification and an L1 loss function as the supplementary loss L_x, in order to retain the content from the original input. Further, we modified the structure of the generator network. By using an encoder and decoder architecture, the generator is able to take a normal photograph as input rather than the 1-dimensional noise in vanilla GAN. We implemented residual blocks to replace the regular convolutional layers in the generator and discriminator networks. This enabled us to build a deeper network and prevented the vanishing and exploding gradients problems of the regular convolutional network during backpropagation and provided a smooth and stable training process. We also explored two embedding approaches in the label embedding layer: the multiplication approach and the linear addition approach. The multiplication approach involves the label more in the style rendering process (Fig 3a), the linear addition approach makes the label contribute less but retains a more clear content (Fig 3b).
This answered our first research question: An improved generative adversarial network structure along with a new loss function combination enables the generation of images with particular artistic styles from normal photographs.

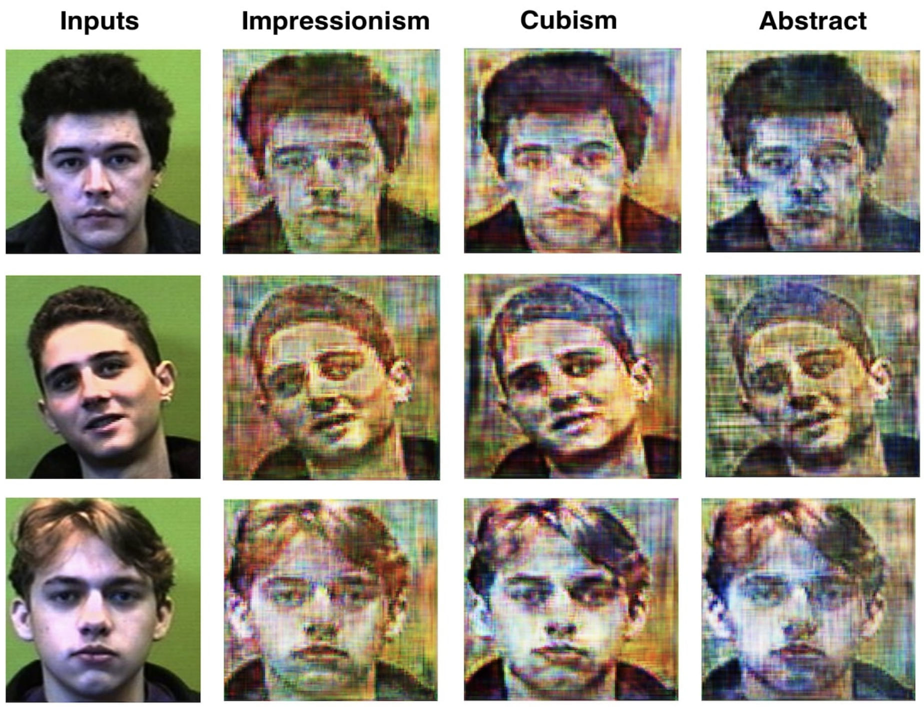


Fig. 10: Sample outputs for $W_y = 2.5, W_c = 1, W_x = 10$

After some fine-tuning, we set W_c to 1 as we found that it will dramatically affect the stability of the outputs and set W_y to 2.5 which is able to bring enough artistic feeling to the outputs. The weights combination $\{W_y = 2.5, W_c = 1, W_x = 10\}$ provides images that meet our personal aesthetic preferences.

Our second research question was:
How can we control the trade-off between style and content of stylized images generated from the above networks?

To answer this question, we explored factors that allow us to control the contributions of loss functions L_y, L_c and L_x, which are loss weights W_x, W_c, and W_y. Increasing a specific weight magnifies the effect of the related loss function, a large value of W_x led to a clear and sharp content but less artistic feeling (Fig 5), a large value of W_c led to intense colours and distinctive styles but made the content blurry (Fig 6), a large value of W_y led to textures similar to the artworks but made the content unrecognizable (Fig 7). We did some further parameter fine-tuning to adjust above three loss weights and obtained results that we found aesthetically pleasing by our personal criteria (Fig 10).

The above work answered our second research question: Adjusting the ratio between the loss weights enables relatively precise control of the trade-off between style and content of a stylized image.

B. Future Work

For future work, first, we hope that we can improve the current label embedding approach in order to generate more refined images. Second, the loss weights could have hundreds of different combinations and it is very difficult to find the combination to please the majority of people in a limited number of experiments. We hope that we can invite audiences from art circles in future work in order to help us to evaluate generated images from a professional perspective. Besides that, the capability of GANs depends not only on the architecture but also on the training data. We would like to explore its potential by using a larger variety of data. Last, to achieve a higher level of creativity, the evolutionary neural network is also a direction we intend to explore. The technique of evolutionary computing may be able to bring more autonomy to generative adversarial networks in order to generate creative paintings without deliberately-prepared input images.

References

[1] L. Comte and Christian, *Argentine Indians*. Consorcio de Editores, 2003.
[2] J. Collomosse and P. Hall, "Genetic paint: A search for salient paintings," in *Applications of Evolutionary Computing*, Rothlauf, Ed. Springer, Berlin, Heidelberg, 2005, pp. 437–447.
[3] J. Romero and P. Machado, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Springer, 11 2007.
[4] G. Hinton, J. McClelland, and D. Rumelhart, *Distributed Representations*, 1984.
[5] H. P. B. L. B. Y. LeCun, Y., "Object recognition with gradient-based learning," in *Feature Grouping*. Springer, 1999.
[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June 2017.

[7] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[9] A. Elgammal, B. Liu, M. Elhoseiny, and M. Mazzone, “CAN: creative adversarial networks, generating “art” by learning about styles and deviating from style norms,” *CoRR*, vol. abs/1706.07068, 2017.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[11] L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style,” *CoRR*, vol. abs/1508.06576, 2015.

[12] F. Luan, S. Paris, E. Shechtman, and K. Bala, “Deep photo style transfer,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[13] X. Mao, Q. Li, H. Xie, Y. Raymond, and Z. Wang, “Multi-class generative adversarial networks with the L2 loss function,” *arXiv*, vol. abs/1611.04076, 2016.

[14] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv*, vol. abs/1511.06434, 2016.

[15] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier gans,” in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML’17. JMLR.org, 2017.

[16] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros, “Context encoders: Feature learning by inpainting,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[17] C. Li and W. Michael, “Precomputed real-time texture synthesis with markovian generative adversarial networks,” in *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016.

[18] P. Isola, J. Zhu, T. Zhou, and A. Efros, “Image-to-image translation with conditional adversarial networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[19] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[20] J. Zhu, T. Park, P. Isola, and A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[21] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” *CoRR*, vol. abs/1607.08022, 2016.